

## CLAIMS

What is claimed is:

- 1     1.     In a computer system having a plurality of threads, including a first and  
2     second thread, a method of executing more than one thread at a time, the method  
3     comprising:  
4         providing a first and a second reorder buffer;  
5         reading first instructions and first operands associated with the first thread  
6     from the first reorder buffer;  
7         executing one of the first instructions and storing a result in the first reorder  
8     buffer, wherein storing the result includes marking the result with a tag associating  
9     the result with the first thread;  
10        reading second instructions and second operands associated with the second  
11    thread from the second reorder buffer; and  
12        executing one of the second instructions and storing a result in the second  
13    reorder buffer, wherein storing the result includes marking the result with a tag  
14    associating the result with the second thread.
  
- 1     2.     The method of claim 1, wherein the method further includes providing a first  
2     execution unit, and wherein executing one of the first or one of the second  
3     instructions includes executing in the first execution unit.
  
- 1     3.     The method of claim 2, wherein the first execution unit includes a floating  
2     point unit.
  
- 1     4.     The method of claim 3, the method further including providing a second  
2     execution unit, and wherein executing one of the first or one of the second  
3     instructions includes executing in the second execution unit.

- 1 5. The method of claim 4, wherein the second execution unit includes a  
2 memory order buffer.
- 1 6. The method of claim 5, the method further including providing an  
2 instruction pipeline.
- 1 7. The method of claim 6, wherein executing one of the second instructions  
2 includes allocating the first and second execution units to the second instruction if  
3 the first thread stalls in the instruction pipeline.
- 1 8. The method of claim 6, wherein executing one of the second instructions  
2 includes allocating the first and second execution units to the second instruction if  
3 the first thread is flushed from the instruction pipeline.
- 1 9. The method of claim 6, wherein the method further includes:  
2 providing a first and second instruction fetch/decode units, wherein the first  
3 instruction fetch/decode unit is associated with the first thread and the second  
4 instruction fetch/decode unit is associated with the second thread; and  
5 spacing the instruction fetch/decode units evenly around the instruction pipeline.
- 1 10. The method of claim 6, wherein executing one of the first or one of the  
2 second instructions includes the first and second threads prioritizing instructions  
3 within its thread.
- 1 11. The method of claim 10, wherein prioritizing instructions includes  
2 increasing a priority of an instruction when the instruction completes a loop around  
3 the instruction pipeline.
- 1 12. The method of claim 10, wherein an instruction is assigned a priority during  
2 compilation based on a dependency of other instructions.

1 13. A processor comprising:  
2 an instruction pipeline; and  
3 a results pipeline, wherein the instruction pipeline and the results pipeline  
4 are counter rotating queues;  
5 a first execution unit in communication with the results and instruction  
6 pipelines;  
7 a plurality of threads including a first and second thread;  
8 a first and a second reorder buffer, the first reorder buffer associated with the  
9 first thread and the second reorder buffer associated with the second thread; and  
10 a first instruction fetch/decode unit.

1 14. The processor of claim 13, further including a second instruction  
2 fetch/decode unit, wherein the first instruction fetch/decode unit is associated with  
3 the first thread and the second instruction fetch/decode unit is associated with the  
4 second thread.

1 15. The processor of claim 14, wherein the first and second fetch/decode units  
2 are spaced evenly at first and second locations around the instruction and results  
3 pipelines.

1 16. The processor of claim 13, wherein instructions are multiplexed from the  
2 first instruction fetch/decode unit into the first and second threads.

1 17. The processor of claim 16, wherein instructions from the first fetch/decode  
2 unit are multiplexed into points spaced evenly at first and second locations around  
3 the instruction and results pipelines.

1 18. The processor of claim 17, further including a second execution unit,  
2 wherein the first and second execution unit are spaced evenly around the instruction  
3 and results pipeline.

1 19. The processor of claim 13, wherein instructions include a tag associating the  
2 instruction with a thread.

1 20. The processor of claim 13, wherein the pipeline is two instructions wide.